

**METHOD OF LOGICAL MODELING OF OPERATOR INTERACTION WITH
PROGRAMMABLE LOGIC CONTROLLER LOGICAL VERIFICATION SYSTEM**

5

CROSS-REFERENCE TO RELATED APPLICATION(S)

The present application claims the priority date of
co-pending United States Provisional Patent Application Serial
Number 60/236,964, filed September 29, 2000.

10

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to
15 programmable logic controllers and, more specifically, to a
method of logical modeling of operator interaction with
programmable logic controller logical verification system for
manufacturing a motor vehicle.

20 2. Description of the Related Art

It is known that programmable logic controller code
is written by controls engineers after assembly tooling designs
are completed and a manufacturing process has been defined.
The creation of the programmable logic controller code is
25 mostly a manual programming task with any automation of the

code generation limited to "cutting and pasting" previously written blocks of code that were applied to similar manufacturing tools. Once the programmable logic controller code is written, it is used by a programmable logic controller
5 integrated on to hard tools in the manufacture of parts for motor vehicles. The programmable logic controller code is not validated (debugged) until the hard tools are built and tried. A significant portion of this tool tryout process is associated with the debugging of the programmable logic controller code at
10 levels of detail from a tool-by-tool level, to a workcell level and finally at a tooling line level.

It is also known that a manufacturing line is typically made of three to twenty linked workcells. Each workcell consists of a fixture to position product (sheet
15 metal) and associated automation (robots) that process the product (welding). The workcell typically consists of a fixture/tool surrounded by three or four robots. A human operator typically interacts with the product in the workcell for the manufacturing line.

20 It is further known that the workcells for a manufacturing line can be modeled before the manufacturing line is implemented. The modeling techniques, such as Robcad from

Tecnomatix and Igrip from Deneb, for the manufacturing process are limited in scope to a workcell level, due to how these type of technologies represent and manipulate three dimensional data and tool motions. It is still further known that RobcadMan
5 from Tecnomatix and Transom Jack from Engineering Animation Inc. are two ergonomic modeling technologies. However, both of these products are focused on visualizing and resolving operator behavior as it applies to "reach" or load conditions and are not suited for modeling operator behavior as it applies
10 to the overall PLC control system design and verification.

To model the proper behavior of a workcell to enable programmable logic controller (PLC) programs to be verified prior to the actual build/launch of tooling, it is necessary to account for the interaction of the operator in the workcell.
15 This interaction consists of two segments: sequential operation where the operator functions as an integral part of the sequential cycle of the workcell, thereby causing certain logic conditions to be set in the PLC logic (ex: loading/unloading a part each cycle); and interrupt or
20 exception behavior where the operator responds to asynchronous requests for the workcell. The premise in building the workcell model for simulation is that a user of a PLC logic

verification system can perform all the necessary asynchronous functions without undue burden (for example, placing the machine into auto cycle). However, it would be difficult to be able to run the workcell through multiple continuous cycles if
5 the user had to interact with the simulation throughout each cycle as if they were the real operator.

Therefore, it is desirable to provide a method for modeling an operator as a logical ingredient in a normal or expected behavior of a workcell. It is also desirable to
10 provide a method for logical modeling of operator interaction with a programmable logic controller logical verification system. It is further desirable to provide a method of logical modeling of operator interaction with a programmable logic controller logical verification system for manufacturing a
15 motor vehicle. Therefore, there is a need in the art to provide a method that meets these desires.

SUMMARY OF THE INVENTION

Accordingly, the present invention is a method of
20 logical modeling operator interaction with a programmable logic controller logical verification system. The method includes the steps of constructing a flowchart of interaction of an

operator in a workcell and testing whether logic of the flowchart is correct. The method also includes the step of using the flowchart to test PLC code to build the workcell if the logic of the flowchart is correct. It should be appreciated that the method may include the ability to force status of the desired operator behavior so that diagnostic conditions can be verified.

One advantage of the present invention is that a method is provided for logical modeling of operator interaction with a programmable logic controller logical verification system. Another advantage of the present invention is that the method allows a user to verify that the PLC code being planned will work as intended, prior to physically building the tools/manufacturing line and locating equipment. Yet another advantage of the present invention is that the method allows verification of PLC code prior to vendor tool tryout (VTTO) and directly shortens the product development timing, resulting in substantial timing and cost savings.

Other features and advantages of the present invention will be readily appreciated, as the same becomes better understood, after reading the subsequent description taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a diagrammatic view of a system, according to the present invention, for using a method of logical modeling of operator interaction with programmable logic controller logical verification system illustrated in operational relationship with an operator.

FIGS. 2 through 5 are flowcharts illustrating a method, according to the present invention, of logical modeling of operator interaction with programmable logic controller logical verification system using the system of FIG. 1.

DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

Referring to the drawings and in particular FIG. 1, one embodiment of a system 10, according to the present invention, is illustrated. In the present invention, a user 12 using a computer 14 logically models operator interaction with a programmable logic controller (PLC) logical verification system 16. The computer 14 sends and receives information with the PLC logical verification system 16 via an electronic link. The PLC logical verification system 16 verifies PLC logic for a workcell of a tooling line. The computer 14 also sends and receives information with an operator interaction design 18 via

an electronic link. The operator interaction design 18 sends and receives information with the PLC logical verification system 16 to verify the PLC code. Once the PLC code is verified, it is exported by the computer 14 via an electronic link to at least one PLC 20. The PLC 20 is then used at physical tool build to produce or build a workcell (not shown), which is used in a tooling line (not shown) for the manufacture of parts (not shown) for a motor vehicle (not shown). It should be appreciated that the computer 14, electronic links, and PLC 20 are conventional and known in the art. It should also be appreciated that one variation that has been proposed it to substitute a computer based emulation of the PLC and having a PLC emulation does not materially affect the above description.

Referring to FIGS. 2 through 5, a method, according to the present invention, for logical modeling of operator interaction with the PLC logical verification system 16 of the system 10 is shown. In general, the user 12 models a human operator (not shown) in context of the part or product being developed, as a controller with a unique set of resources. In the present invention, a controller may be an operator, robot, PLC, or any programmable device. The resource assigned to the

operator controller is a model of a physical manifestation of the operator in the workcell. For example, an operator resource for moving a part in a repetitive cycle might be an overhead crane that they directly control. Once the resource
5 is bound to the operator controller, the PLC code or controller program can be written using conventional logic.

The method begins by writing a control model for an operator by the operator interaction design system 18. For example, the operator interaction design system 18 will create
10 a control model definition that describes how an operator picks up a part, carries, and loads it into a fixture. As illustrated in FIG. 5, a flowchart of a method for a part flow in two stations or workcells is shown. The method begins in bubble 50 where an operator or pusher moves the part to Station
15 0 and on a transfer bar in block 52. The part moves forward in block 54, moves up in block 56, until the part is at a top in block 58. The part drops to the transfer bar in block 60. The part is at workspace 1 in block 62. The part moves forward to Station 1 in block 64. The part moves up in block 66, until
20 the part is at the top in block 68. The part drops to a second transfer bar in block 70. The part is at workspace 2 in block 72. It should be appreciated that the control model is

information that describes events, dependencies, and logical conditions. It should also be appreciated that the method uses flowcharts to represent the cyclic logical behavior of the operator. It should further be appreciated that the purpose of the model is to verify the PLC logic by providing input signals to the PLC at desired times or based on conditional events. It should also be appreciated that the focus is on the logical representation of the operator and not the visual or spatial representations.

10 Referring to FIG. 2, the method begins in bubble 100 and advances to block 102. In block 102, the method includes selecting "commands" from a resource's capability to cause an action. The commands available to the operator flowchart (selected while the user 12 is in the action block) are
15 determined by the collection of resources that have been bound to that controller. For example, a resource may be "Carry" for an operator to carry a workpiece from a first location to a second location. The resource has at least one, preferably a plurality of capabilities. For example, a capability for the
20 resource "carry" may be "lift" such that the operator lifts the workpiece before carrying the workpiece from the first location to the second location. From block 102, the method advances to

diamond 104 and determines whether the user 12 is done. If so, the method advances to bubble 106 and ends. If not, the method advances to block 102 previously described. It should be appreciated that the operator controller has at least one resource and the at least one resource has at least one capability. It should also be appreciated that the method is carried out on the computer 14 by the user 12.

Referring to FIG. 3., the method allows either the operator logic or PLC to test for status being returned from a resource through its input registers. The execution of the flowchart during simulation, which might be more than one based on different starting conditions, then proceeds by successive "command, test status" loops. The method begins with initializing the test in bubble 200. From bubble 200, the method advances to block 202 and idles the operator. For example, in block 202, the operator is set to idle where no work or motion is being performed. From block 202, the method advances to block 204 and starts a timer. The timer may be set for a predetermined time period such as ten seconds to carry out the commands as illustrated in FIG. 4. The user 12 receives verification from the system 10 when the commands are completely performed. The user 12 determines whether the

commands are completed within the predetermined time period. After block 204, the method advances to bubble 206 and ends. It should be appreciated that branching opportunities, reflecting testing multiple possibilities based on various status conditions, is also implemented. It should also be appreciated that controllers test conditions from resources or locations. It should be appreciated that the user can force conditions in the operator logic that allows diagnostic conditions to be verified.

10 Referring to FIG. 4, the method executes the commands when the timer is started. The method begins in bubble 300 on the timer being started when called by block 204 of FIG. 3. From bubble 300, the method advances to block 302 and the operator gets, for example, a woodblock, which is a part. The method advances to block 304 and the operator takes the woodblock from a part source. The method advances to block 306 and commands the operator to put the woodblock. The method advances to block 308 and the operator puts the woodblock at a first part location. The method advances to block 310 and commands the operator to push "cyclestart". The method advances to bubble 312 and ends. It should be appreciated that the flowchart evokes resources and the user 12 can test the

flowchart as to what is being tested is a PLC program loaded to the controller. It should also be appreciated the specific sequence of commands in FIG. 4 is merely an example of the commands executed.

5 The present invention has been described in an illustrative manner. It is to be understood that the terminology, which has been used, is intended to be in the nature of words of description rather than of limitation.

10 Many modifications and variations of the present invention are possible in light of the above teachings. Therefore, within the scope of the appended claims, the present invention may be practiced other than as specifically described.